In the Claims:

Claim 1 (currently amended): A computer implemented method for simulating a system design containing at least two components, said method comprising:

identifying said components:

creating models of said components in a high level general purpose programming language;

creating a set of function calls in a high level general purpose programming language, wherein at least one of said function calls comprises a cycle-based function corresponding to a collection of communication events;

combining said models to form ~~said~~ a virtual prototype;

linking one of said models with another of said models using said set of function calls;

executing said virtual prototype, wherein said models communicate and cycle accurate information is generated.

Claim 2 (original): The computer implemented method of claim 1 wherein said creating models of said components step further comprises:

creating a blank component model;

adding a sub-component to said blank component model; and

configuring one or more parameters for said sub-component.

Claim 3 (original): The computer implemented method of claim 2 wherein at least two of said sub-components are added to said blank component model, further comprising:

linking said sub-components with said set of function calls.

Claim 4 (original): The computer implemented method of claim 2 wherein said components are added into a component repository after said components are created.

Claim 5 (original): The computer implemented method of claim I wherein said set of function calls comprises:

a control interface for communications between the simulation environment and said components; and

a peer interface for communications between said components.

Claim 6 (original): The computer implemented method of claim 5 wherein said peer interface comprises clock functions, access functions, and signal functions.

Claim 7 (original): The computer implemented method of claim 1 wherein said combining step further comprises:

loading a component from a component repository;

identifying sub-components of said component; and

elaborating said sub-components.

Claim 8 (original): The computer implemented method of claim 7 wherein said elaborating step further comprises:

instantiating said sub-components;

configuring said sub-components; and

linking said sub-components with said set of function calls.

Claim 9 (original): The computer implemented method of claim 1 wherein said generating cycle accurate information step further comprises:

dividing activities in said simulation environment into a first plurality of activities comprising an execute phase and a second plurality of activities comprising an update phase;

computing said first plurality of activities comprising said execute phase at a clock edge;

updating at said clock edge a state of said simulation environment; and

computing said second plurality of activities comprising said update phase at said clock edge.

Claim 10 (previously presented): A computer implemented method for simulating a digital system design in a cycle based simulation environment, comprising:

creating a system design model in a high level general purpose programming language comprising at least two components;

creating a software interface in a high level general purpose programming language that is available to said at least two components, wherein said software interface includes a cycle-based function corresponding to a collection of communication events;

executing said system design model, wherein said at least two components communicate via said interface; and

maintaining cycle accurate information during the simulation.

Claim 11 (original): The computer implemented method of claim 10 wherein said creating a system design model step further comprises:

creating a blank component model;

adding a sub-component to said blank component model; and

configuring one or more parameters for said sub-component.

Claim 12 (original): The computer implemented method of claim 11 wherein at least two of said sub-components are added to said blank component model, further comprising:

linking said sub-components with said set of function calls.

Claim 13 (original): The computer implemented method of claim 11 wherein said components are added into a component repository after said components are created.

Claim 14 (original): The computer implemented method of claim 12 wherein said set of function calls comprises:

a control interface for communications between said simulation environment and said components; and

a peer interface for communications between said components and between said sub- components.

Claim 15 (original): The computer implemented method of claim 14 wherein said peer interface comprises clock functions, access functions, and signal functions.

Claim 16 (original): The computer implemented method of claim 10 wherein said maintaining cycle accurate information step further comprises:

dividing simulation activities in said simulation environment into a first plurality of activities comprising an execute phase and a second plurality of activities comprising an update phase;

computing said first plurality of activities comprising said execute phase at a clock edge;

updating at said clock edge a state of said simulation environment; and

computing said second plurality of activities comprising said update phase at said clock edge.

Claim 17 (previously presented): A computer program product embodied on a computer-readable medium, which when executed, causes a processing system to simulate a system design within a cycle based simulation environment, said computer program product comprising:

instructions for identifying at least two components in said system design;

instructions for integrating models of said at least two components, wherein said models are created in a high level general purpose programming language;

instructions for linking said models using a set of function calls created in a high level general purpose programming language, wherein at least one of said set of function calls comprises a cycle-based function corresponding to a collection of communication events; and

instructions for executing said system design, wherein said models communicate through said link and cycle accurate information is generated.

Claim 18 (original): The computer program product of claim 17 wherein said instructions for creating models further comprises:

instructions for creating a blank component model;

instructions for adding at least one sub-component to said blank component model;

instructions for configuring one or more parameters for said sub-component; and

instructions for linking said sub-components using said set of function calls.

Claim 19 (original): The computer program product of claim 18 further comprising instructions for adding said models of said components into a component repository after said models are created.

Claim 20 (original): The computer program product of claim 18 wherein said instructions for creating a set of function calls comprise:

instructions for creating a control interface for communications between said simulation environment and said components; and

instructions for creating a peer interface for communications between said components and between said sub-components.

Claim 21 (original): The computer program product of claim 20 wherein said instructions for creating a peer interface further comprise:

instructions for creating clock functions;

instructions for creating access functions; and

instructions for creating signal functions.

Claim 22 (original): The computer program product of claim 17 wherein said instructions for generating cycle accurate information further comprise:

instructions for dividing simulation activities in said simulation environment into a first plurality of activities comprising an execute phase and a second plurality of activities comprising an update phase;

instructions for computing said first plurality of activities comprising said execute phase at a clock edge;

instructions for updating at said clock edge a state of said simulation environment; and

instructions for computing said second plurality of activities comprising said update phase at said clock edge.

Claim 23 (previously presented): A method for simulating a design containing at least two components, said method comprising:

creating a model representing each of said at least two components, wherein said models correspond via at least one function call and comprise a virtual prototype, wherein said at least one function call comprises a cycle-based function corresponding to a collection of communication events; and

executing said virtual prototype to generate cycle accurate information.

Claim 24 (original): The method of claim 23 wherein said creating step comprises creating said model in a high level general purpose programming language.

Claim 25 (original): The method of claim 23 wherein said creating models of said components step further comprises:

creating a blank component model;

adding a sub-component to said blank component model; and

configuring one or more parameters for said sub-component.

Claim 26 (original): The method of claim 25 wherein at least two of said sub-components are added to said blank component model, further comprising:

linking said sub-components with said at least one function call.

Claim 27 (original): The method of claim 25 wherein said components are added into a component repository after said components are created.

Claim 28 (original): The method of claim 23 wherein said executing step is performed in a cycle based simulation environment, said method further comprising:

creating a set of function calls in a high level general purpose programming language, wherein said set of function calls comprises:

a control interface for communications between said cycle based simulation environment and said components; and

a peer interface for communications between said components.

Claim 29 (original): The method of claim 28 wherein said peer interface comprises clock functions, access functions, and signal functions.

Claim 30 (original): The method of claim 23 wherein said virtual prototype is created by steps comprising:

loading a component from a component repository;

identifying sub-components of said component; and

elaborating said sub-components.

Claim 31 (original): The method of claim 30 wherein said elaborating step further comprises:

instantiating said sub-components;

configuring said sub-components; and

linking said sub-components with said set of function calls.

Claim 32 (original): The method of claim 23 wherein said executing step is performed in a cycle based simulation environment and said generating cycle accurate information step further comprises:

dividing activities in said cycle based simulation environment into a first plurality of activities comprising an execute phase and a second plurality of activities comprising an update phase;

computing said first plurality of activities comprising said execute phase at a clock edge;

updating at said clock edge a state of said simulation environment; and

computing said second plurality of activities comprising said update phase at said clock edge.

Claim 33 (previously presented): A computer program product embodied on a computer-readable medium, which when executed, causes a processing system to simulate a system design containing at least two components, said computer program product comprising:

instructions for integrating a model representing each of said at least two components, wherein said models communicate via at least one function call and comprise a virtual prototype, wherein said at least one function call comprises a cycle-based function corresponding to a collection of communication events; and

instructions for executing said virtual prototype to generate cycle accurate information.

Claim 34 (original): The computer program product of claim 33 wherein said models are created in a high level general purpose programming language.

Claim 35 (original): The computer program product of claim 33 wherein said instructions for executing said virtual prototype to generate cycle accurate information further comprise:

instructions for dividing simulation activities in said simulation environment into a first plurality of activities comprising an execute phase and a second plurality of activities comprising an update phase;

instructions for computing said first plurality of activities comprising said execute phase at a clock edge;

instructions for updating at said clock edge a state of said simulation environment; and

instructions for computing said second plurality of activities comprising said update phase at said clock edge.